

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**  
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:	§	Group Art Unit: 2151
	§	
Sai V. Allavarpu, et al.	§	Examiner: Dinh, Khanh Q.
	§	
	§	Atty. Dkt. No.: 5181-46200
	§	P4466
Serial No. 09/552,985	§	
	§	
	§	
Filed: April 21, 2000	§	
	§	
For: CORBA Metadata Gateway to	§	
Telecommunications	§	
Management Network	§	

**APPEAL BRIEF**

**Mail Stop Appeal Brief - Patents**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir/Madam:

In response to the Notice of Panel Decision mailed November 13, 2007, Appellants present this Appeal Brief. **Since Appellants' previous appeal (Brief filed December 28, 2004) was not heard by the Board, no fee should be due for this Appeal Brief (see *M.P.E.P.* § 1207.04).** Also, no extension of time is due since this Appeal Brief is filed within one month of the mailing date of the Notice of Panel Decision. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

**I. REAL PARTY IN INTEREST**

As evidenced by the assignment recorded at Reel 010993, Frame 0870, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

## **II. RELATED APPEALS AND INTERFERENCES**

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

### **III. STATUS OF CLAIMS**

Claims 1-31 are pending and rejected. The rejection of claims 1-31 is being appealed. A copy of claims 1-31 as on appeal is included in the Claims Appendix hereto.

#### **IV. STATUS OF AMENDMENTS**

No amendments have been submitted subsequent to the final rejection.

## **V. SUMMARY OF CLAIMED SUBJECT MATTER**

Independent claim 1 is directed to a method for managing a network using a metadata gateway. A client (*See, e.g.*, FIGs. 1a, 2 and 3, items 171a – 171n and 206a - 206b) may generate a request for type information (*See, e.g.*, FIG. 2, item 214a – 214b and FIG. 6, item 612) for an attribute or event pertaining to management of one or more managed network objects (*See, e.g.*, FIG. 2, item 212; page 8, lines 15-21, page 17, line 27 – page 18, line 21). The request may be expressed in an interface definition language operable to define object interfaces across a plurality of platforms and across a plurality of programming language (*See, e.g.*, FIG. 2, item 214a – 214b). Each managed network object is a computer programming language object representing one or more devices on a network (*See, e.g.*, FIGs. 1a and 2, items 150, 151, 152, 153, 154, 155 and 156).

The request of type information may be sent to an object request broker (*See, e.g.*, FIGs. 2 and 3, item 202 and page 16, line 21 – page 17, line 6). Metadata may be retrieved through the metadata gateway (*See, e.g.*, FIG. 3, item 320) by a client manager application sending a request for type information for a managed object attribute or event in IDL through a CORBA Object Request Broker (ORB) (*See, e.g.*, FIGs. 2 and 3, item 202) to the metadata gateway, which then reads the type information from a metadata repository (*See, e.g.*, FIG. 3, item 322), where the type information is stored in a database format. For example, as described at page 16, line 21 – page 17, line 23 of the Specification, manager applications 206 may be communicably coupled to a CORBA Object Request Broker (ORB) and may send IDL request and receive IDL responses through the ORB (see also, FIG. 6, page 23, lines 19-30, page 10, line 14-22).

The metadata gateway may receive the request from type information from the object request broker. The metadata gateway provides translation of metadata to and from a database format and Interface Definition Language (IDL), which is operable across a plurality of platforms and across a plurality of programming languages (*See, e.g.*, FIG. 3, page 8, lines 15-21, page 17, line 27 – page 18, line 21). Type information may be read from a metadata repository storing the type information in a database format may

be translated from the database format to the interface definition language. (*See, e.g.*, page 8, lines 3-13, page 18, line 5 – 21, page 19, lines 11-27). The metadata gateway then translates the retrieved type information from the database format to IDL and sends the translated type information to the ORB, which sends the translated type information for the attribute or event to the client manager application. The metadata gateway may then send the translated type information to the object request broker and the client may receive the translated type information, expressed in the interface definition language, for the attribute or event through the object request broker. (*See, e.g.*, FIG. 6, page 23, line 19- page 24, line 2).

Independent claim 10 is directed to a method or managing a network. As recited in claim 10, a client (*See, e.g.*, FIGs. 1a, 2 and 3, items 171a – 171n and 206a - 206b) may generate a request to encode type information (*See, e.g.*, FIG. 2, item 214a – 214b and FIG. 6, item 612) for an object, attribute, or event, pertaining to one or more managed network objects(*See, e.g.*, FIG. 2, item 212; page 8, lines 15-21, page 17, line 27 – page 18, line 21). The request is expressed in an interface definition language, wherein the interface definition language is operable to define object interfaces across a plurality of platforms and across a plurality of programming languages (*See, e.g.*, page 10, line 25 – page 11, line 2, page 18, lines 11-21, FIG. 4, page 20, line 23-26, FIG. 7, page 24, lines 6-13).

Claim 10 further recites sending the request to an object request broker (*See, e.g.*, FIGs. 2 and 3, item 202 and page 16, line 21 – page 17, line 6) and a metadata gateway (*See, e.g.*, FIG. 3, item 320) receiving the request to encode the type information from the object request broker (*See, e.g.*, FIG. 3, page 17, line 29 – page 18, line 9). Additionally, the type information may be translated from the interface definition language to a database format and may be stored in a metadata repository in a database format (*See, e.g.*, FIG. 7, page 18, lines 11-21, page 24, line 21-29). Thus, metadata may be encoded through the metadata gateway by sending the metadata in IDL to the metadata gateway, which translates the type information from IDL to a database format and stores the type information in the metadata repository.

Independent claim 14 is directed to a network management system comprising a metadata repository and a metadata gateway. The metadata repository may include metadata, expressed in a database format, concerning object classes for managed objects (*See, e.g.*, FIG. 2, item 212; page 8, lines 15-21, page 17, line 27 – page 18, line 21) that are computer programming language objects corresponding to managed devices on a network (*See, e.g.*, page 8, lines 9-13, page 23, lines 20-12). The metadata gateway may be coupled to the metadata repository and to an object request broker (*See, e.g.*, FIG. 6, page 23, lines 19-30, page 10, line 14- 22, 16, line 21 – page 17, line 23).

The metadata gateway may send and receive metadata from the database and may provide translation of the metadata to and from the database format and an interface definition language operable to define object interfaces across a plurality of platforms and across a plurality of programming languages (*See, e.g.*, page 8, lines 3-13, page 18, line 5 – 21, page 19, lines 11-27, FIG. 7, page 18, lines 11-21, page 24, line 21-29).

Independent claim 22 is directed to a computer-readable storage medium comprising program instructions that are computer-executable to implement a metadata gateway (*See, e.g.*, FIG. 3, item 320, page 10, line 25 – page 11, line 2, page 18, lines 11-21) receiving a request for type information from an object request broker (*See, e.g.*, FIGs. 2 and 3, item 202 and page 16, line 21 – page 17, line 6) where the type information pertains to management of one or more managed network objects (*See, e.g.*, the discussion above regarding claim 1, FIGs. 3 and 6, page 8, lines 15-21, page 17, line 27 – page 18, line 21, and page 23, lines 19-30). The program instructions also implement reading the type information from a metadata repository, wherein the type information is stored in a database format in the metadata repository (*See, e.g.*, FIG. 3, item 322, page 8, line 7 – page 9, line 3, page 10, line 14- 22). Claim 22 further recites translating the type information from the database format to an interface definition language, and the metadata gateway sending the translated type information to the object request broker (*see, e.g.*, page 18, line 11-21, page 19, line 18-27). Further discussion



and example embodiments regarding the limitations of claim 22 are discussed above regarding claim 1.

Independent claim 27 is directed to a computer-readable storage medium comprising program instructions that are computer-executable to implement a metadata gateway (*See, e.g.*, FIG. 3, item 320, page 10, line 25 – page 11, line 2, page 18, lines 11-21) receiving a request to encode type information from an object request broker (*See, e.g.*, FIGs. 2 and 3, item 202 and page 16, line 21 – page 17, line 6) where the type information pertains to management of one or more managed network objects (*see, e.g.*, page 10, line 25 – page 11, line 2, page 18, lines 11-21, FIG. 4, page 20, line 23-26, FIG. 7, page 24, lines 6-13). The program instructions are also computer-executable to implement translating the type information from an interface definition language to a database format and storing the type information in a metadata repository (*See, e.g.*, FIG. 3, item 322 and page 17, line 29 – page 18) using a database format (*See, e.g.*, FIG. 3, line 9, FIG. 7, page 18, lines 11-21, page 24, line 21-29).

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

**VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

1. Claims 1-5, 7-12, 14 and 17-31 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Carre (U.S. Patent 6,282,579) in view of Hamilton et al. (U.S. Patent 5,758,186, hereinafter "Hamilton").

2. Claims 6, 13, 15 and 16 stand finally rejected 35 U.S.C. § 103(a) as being unpatentable over Carre in view of Hamilton and further in view of Kung et al. (U.S. Patent 6,775,267) (hereinafter "Kung").

## VII. ARGUMENT

### First Ground of Rejection:

Claims 1-5, 7-12, 14 and 17-31 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Carre (U.S. Patent 6,282,579) in view of Hamilton et al. (U.S. Patent 5,758,186, hereinafter “Hamilton”). Appellants respectfully traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

### Claims 1, 5, 7 and 9:

1. **The cited art fails to teach or suggest a client generating a request for type information for an attribute or event pertaining to management of one or more managed network objects.**

In regards to claim 1, contrary to the Examiner’s contention, **Carre in view of Hamilton fails to teach or suggest a client generating a request for type information for an attribute or event pertaining to management of one or more managed network objects, wherein each managed network object is a computer programming language object representing one or more devices on a network.** As argued previously, the Examiner erroneously equates *any address conversion* performed as part of any remote procedure call with specifically *a client generating a request for type information* for an attribute or event pertaining to management of one or more managed network objects. However, as shown below, merely performing an address conversion does not teach or suggest *a client generating a request for type information*.

Carre pertains to address conversion between CORBA objects and OSI objects (Carre - col. 1, lines 9-19; col. 1, line 59 - col. 2, line 46) and to transforming of object interfaces (col. 5, lines 49-59), while Hamilton teaches a system for handling diverse protocols with remote procedure calls (Abstract, col. 1, line 57 – col. 2, line 34). The

Examiner cites the Abstract, FIG. 2A, 2B, as well as column 3, line 18 to column 4, line 62 and column 6, lines 10-35 of Carre regarding a client generating a request for type information for an attribute or event.

Carre is concerned with *converting address types and object interfaces*, but fails to teach anything regarding *a client generating a request for type information*. At the Examiner's cited passages, Carre describes how objects interact and communicate using a CORBA infrastructure. Carre teaches that his system converts from one address type to a corresponding type of another specification language or address mode, such as to permit interaction between two objects that use different address modes. See, Carre, column 1, line 59 - column 2, line 6; column 2, lines 11-14; lines 25-28; and lines 34- 39.

Even if combined with Hamilton, Carre's system does not include a client *generating a request for type information*, either at the Examiner's cited passages or elsewhere.

In Response, the Examiner relies on Carre's teachings regarding "communicating via the Network Management Protocol CMIP, wherein the OSI objects including different specification languages such as CORBA IDL, ASN.1" and repeating the citations from the rejection of claim 1 (i.e., abstract, FIGs 2a and 3a, col. 3, line 18 – col. 5, line 62 and col. 6 lines 10-35). However, as noted above, Carre, at the cited passages and elsewhere, teaches general address conversion as part of remote procedure calls when two objects, esp. those with different addressing modes, communicate remotely.

The Examiner apparently equates any address conversion that may occur during the course of a remote procedure as *a client specifically generating a request* for type information. However, the Examiner's interpretation is incorrect. Address conversions that are made during the normal course of other processing do not require *a client generating a request for type information*. For instance, Carre's system does not include any such request being generated.

Carre in view of Hamilton teaches address conversion, but fails to teach or suggest a client generating a request for type information. Converting address types and object interfaces as part of a protocol conversion, as taught by Carre in view of Hamilton, does not include *a client generating a request* for type information. The Examiner has never explained how or when or why the client in Carre actually generates a request for type information. In fact, as discussed below, the purpose of Carre's teachings is for any required conversion to be performed transparent to the client. A client request for type information would make no sense in Carre's system and would be counter to the intended operation of Carre's system.

Carre teaches, even when combined with Hamilton, that client objects request services provided by service objects. Carre further teaches that a client object sends a request message to a service object that contains an operation, a target object, one or more parameters, and, optionally, a request context (Carre, column 3, lines 47-53). **However, Clients in Carre's system do not generate requests for type information,** but instead request a specific service from a particular server object. Carre's system may automatically make any type and address conversions between different object specification languages necessarily to allow the client and server object to communicate. Thus, in a system resulting from the Examiner's combination of Carre and Hamilton, type and address conversion may take place, **without the agent (client) knowing** that such conversions are performed on the parameters, data, and addresses of various requests between Carre's manager and agent objects. The clients in Carre do not have a need of, nor generate requests for, type information. Instead, Carre's system is configured to allow clients to request services from servers without having to perform any type conversion themselves. Thus, clients in Carre have no need to generate requests for type information.

The address conversion according to Carre and Hamilton takes place without the client even knowing that such a conversion is taking place and cannot be considered to include the client generating a request for type information.

**2. The cited art teaches away from a client generating a request for type information for an attribute or event.**

Moreover, there is no need in Carre's system for a *client* to generate a request for type information for an attribute or event. In fact, the purpose of Carre's teachings appears to be to perform the address type conversion for the client **so that the client does not have to modify its own interface in order to communicate with an object employing a different addressing mode**. See, e.g., col. 1, line 43 – col. 2, line 6. **Therefore, Carre actually teaches away from a client generating a request for type information for an attribute or event.** For a client to have to request type information would be counter to the intended operation of Carre's teachings. "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959).

**3. The cited art fails to teach or suggest sending the request for type information generated by a client to an object request broker.**

In further regard to claim 1, **Carre in view of Hamilton also fails to disclose sending the request for type information generated by a client to an object request broker.** The Examiner admits that Carre fails to teach or suggest sending a request for type information to an object request broker and relies on Hamilton, citing the abstract, Fig. 1, col. 3, lines 4 – 67 and col. 4, lines 14 – 60). However, like Carre, Hamilton does not teach or suggest anything about sending a *client request for type information* to an object request broker.

Hamilton teaches a system for handling diverse protocols with remote procedure calls. Specifically, Hamilton teaches that a server receives protocol-dependent method calls that are specified in a protocol-dependent format. The protocol-dependent method calls in Hamilton include method descriptors that are compared to other protocol-

dependent values. Matching values, from the comparison, are passed to a protocol-independent processing module that executes the method and returns the reply to the client. See, e.g., Hamilton, column 1, line 57-column 2, line 34. Thus, Hamilton, whether considered singly or in combination with Carre, teaches remote procedure calls using method calls that may include an address conversion. However, such remote procedure calls, even if using in-line address conversions, are not, and cannot be considered to teach or suggest the specific limitation of sending the request for type information that was generated by a client to an object request broker, as recited in Appellant's claim.

In response, the Examiner merely refers to Hamilton's subcontract server 58 "performing data marshalling and other operations of method invocations" (Final Action, pp. 14-15). "Performing data marshalling and other operations of method invocations" does not teach or suggest the specific limitations of Appellant's claim. The Examiner also refers to "managing operations such as OSI objects on the Agent upon requests from the Manager unit" and cites item M of Fig. 2a from Carre. However, Carre, even if combined with Hamilton, does not describe his manager unit as sending (or receiving) a request (generated by a client) for type information.

As noted previously, the Examiner is erroneously equating the automatic conversions performed as part of a protocol translation with the specific limitation of a client generating a request for type information and that client request being sent to an object request broker.

**4. The cited art fails to teach or suggest the client receiving the translated type information.**

**In further regard to claim 1, Carre in view of Hamilton also fails to teach or suggest the client receiving the translated type information.** Claim 1 requires, among other things, that the client generate a request for type information for an attribute or event, and that the client receive the translated type information for the attribute or event.

Following the Examiner's line of reasoning, agent (A) or manager unit (U) would have to receive the requested type information, translated to IDL. However, the automatic address and/or type conversions performed as part of Carre's protocol translation are performed on communication messages sent between Carre's agent and manager units.

Carre performs automatic conversions so that an object configured to communicate via one protocol can understand messages in a different protocol from another object. However, **any address or other information converted or translated by Carre's system is not returned to the object or unit sending the request.**

In Carre, the converted data, parameters and address are sent to the receiving object. Carre's client cannot be considered to receive translated type information in light of the fact that Carre's address conversions are performed on the communication messages sent by the client to another object, such that any translated information is sent on to the receiving object, not returned to the sending object. Furthermore, any return communication from the receiving object back to the requesting object would be subject to Carre's protocol translation to convert any data into the protocol supported by the requesting object. In other words if Carre's system performs protocol translation in the service request communications, such as from protocol A to protocol B, Carre's system would then perform the reverse conversion (e.g., from B to A) on the response, since the original requesting object supports protocol A.

Thus, the converted address and/or other information converted as part of delivering a request from Carre's managing object (M) to Carre's agent object (A) is not then returned to the managing object. Instead, a message sent from M to A in Carre's system would be converted as part of Carre's protocol translation and then sent to A. The converted message is not returned to M. Instead, a separate, return, message from A may be sent back to M and may also be translated to a different protocol. Thus, Carre's object communication messages, even if combined with Hamilton, cannot be considered request for type information or the client receiving the translated type information, as recited in claim 1.



**5. The cited art fails to teach or suggest a metadata gateway receiving the request for type information from the object request broker.**

In further regard to claim 1, **Carre in view of Hamilton further fails to disclose a metadata gateway receiving the request for type information from the object request broker**. The Examiner admits that Carre fails to teach or suggest a metadata gateway receiving the request for type information from the object request broker and relies on Hamilton. However, Hamilton, even when combined with Carre, fails to teach or suggest anything regarding a metadata gateway *receiving a request for type information* from an object request broker.

The Examiner, in the response to arguments, again cites Fig. 1, column 3, lines 4 – 67 and column 4, lines 14 – 60 of Hamilton. However the Examiner's reliance on these passages, an on Hamilton in general, regarding this limitation of claim 1 is misplaced. The cited passages describe how Hamilton's client stubs pass remote procedure calls to client subcontract processes that, in turn, package the remote procedure calls for transport. On the server side, according to Hamilton, server subcontract processes passes the received package to server stubs that make procedure calls "to execute a specified method of the invoked object."

Thus, the cited portions of Hamilton describe a remote procedure call mechanism. The cited portions do not describe anything regarding a request for type information or about a metadata gateway receiving a request for type information from an object request broker. As noted above, a remote procedure call mechanism, even if it includes automatic type conversion is not a request for type information, and is clearly not a request for type information received by a metadata gateway from an object request broker. In fact, nowhere does Hamilton mention anything regarding a receiving a request for type information from an object request broker. Thus, contrary to the Examiner's assertion, Hamilton combined with Carre does not teach or suggest **a metadata gateway receiving the request for type information from the object request broker**.

**6. The cited art fails to teach or suggest reading the type information from a metadata repository, where the type information is stored in a database format.**

Contrary to the Examiner's contention, **Carre in view of Hamilton does not teach or suggest reading the type information from a metadata repository, wherein the type information is stored in a database format in the metadata repository.** The Examiner cites the CMISE/IDL in FIG. 3 of Carre. However, the cited CMISE/IDL is not a metadata repository from which type information is read. The Examiner refers to using CMISE/IDL from FIG. 3 as a metadata repository to manage and/or store translated OSI objects. However, Carre does not teach or suggest using the CMISE/IDL of FIG. 3 as a metadata repository to manage or store translated OSI objects.

In contrast, Carre teaches that the CMISE/IDL is an "interface unit" that contains the "CMISE object and the services assigned to this object" (Carre, column 5, lines 21-39). Carre also teaches that the CMISE object is specified by an IDL interface and acts, and thus appears, like a CORBA object. Thus, the CMISE/IDL cited by the Examiner is a communication interface that allows non-CORBA objects to be interacted with via standard CORBA interfaces and clearly not a metadata repository in which type information is stored in a database format.

Moreover, Appellants' claim specifically recites that *type information* is stored in *a database format* in the metadata repository. Carre's gateway does not store type information in a database format and thus cannot be considered the metadata repository of Appellants' claim.

**7. The cited art fails to teach or suggest the client receiving the translated type information for the attribute or event through the object request broker.**

**Carre in view of Hamilton also fails to teach or suggest the client receiving the translated type information for the attribute or event through the object request broker.** The Examiner's cited portion of Carre (i.e., col. 6, lines 10-35) does not describe a client receiving the translated type information for the attribute or event through the object request broker. Instead, as noted above, the cited passage is part of a description of converting objects between ASN and IDL, as well as caching of converted object instances. As described above, converting CMISE and CORBA objects is not the same as client *generating a request for type information, and receiving the translated type information for the attribute or event through an object request broker.* The Examiner has overlooked the fact that Carre makes no mention, neither in the cited passage nor elsewhere, of any client receiving translated type information for an attribute or event through an object request broker. Without some teaching from the cited art, the Examiner's assertion amounts to nothing more than opinion and speculation, neither of which are proper standards for rejection.

Moreover, clients in Carre's system, even if combined with Hamilton's teachings, do not receive any translated type information through an object request broker. Carre and Hamilton both teach remote procedure call mechanism for executing remote methods over a network. Consequently, clients in Carre's and Hamilton's systems, whether considered singly or in combination, receive the results of the executed method. Clients in Carre and Hamilton do not receive translated type information through an object request broker.

### **Claims 2 and 3:**

**1. The cited art fails to teach or suggest translating the type information from the database format to an abstract syntax notation and translating the type information from the abstract syntax notation to the interface definition language.**

Regarding claim 2, contrary to the Examiner's assertion, Carre in view of Hamilton does not teach or suggest **translating the type information from the**

**database format to an abstract syntax notation and translating the type information from the abstract syntax notation to the interface definition language.** The Examiner cites column 1, lines 34-55 and column 5, line 60 – column 6, line 21, specifically referring to Carre's teachings regarding using semantic conversions. However, the cited passages, as well as the remainder of Carre, even if combined with Hamilton, only describe converting address values from abstract syntax notation to interface definition language as part of executing a remote procedure call.

For example, Carre teaches converting an ASN-specified OSI address value to a correspondent IDL type (Carre, col. 1, lines 40-62 and col. 6, lines 1-2). Thus, the Examiner's cited passages describe converting between ASN and IDL (and the reverse), but fail to mention anything about translating *from a database format* to an abstract syntax notation.

Additionally, Hamilton, even if combined with Carre, also fails to teach or suggest translating the type information from the database format to an abstract syntax notation and translating the type information from the abstract syntax notation to the interface definition language. Thus, Carre and Hamilton, whether considered singly or in combination, do not teach or suggest the limitations of claim 2.

**Claim 4:**

**The cited art fails to teach or suggest translating the type information from the abstract syntax notation to an object specification language; and translating the type information from the object specification language to the interface definition language.**

Regarding claim 4, contrary to the Examiner's assertion, Carre in view of Hamilton does not teach or suggest **translating the type information from the abstract syntax notation to an object specification language and translating the type information from the object specification language to the interface definition**

language. The Examiner cites column 1, lines 34-55 and column 5, line 60 – column 6, line 21, specifically referring to Carre’s teachings regarding using semantic conversions. However, as noted above regarding claim 2, the cited passages, as well as the remainder of Carre, even if combined with Hamilton, only describe converting address values from abstract syntax notation to interface definition language as part of executing a remote procedure call.

For example, Carre teaches that an OSI address value is converted to a correspondent IDL type (Carre, column 6, lines 1-2). Please note that Carre teaches that OSI objects are specified in ASN (Carre, column 1, lines 40-42). Thus, the Examiner’s cited passages describe converting directly between ASN and IDL (and the reverse), but fail to mention anything about translating from an abstract syntax notation to an object specification language and converting the object specification language to the IDL.

Additionally, Hamilton, even if combined with Carre, also fails to teach or suggest translating the type information from the abstract syntax notation to the object specification language and translating the type information from the object specification language to the interface definition language. Thus, Carre and Hamilton, whether considered singly or in combination, do not teach or suggest the limitations of claim 4.

**Claim 8:**

**The cited art fails to teach or suggest wherein the metadata gateway is distributed over a plurality of servers, where each of the plurality of servers presents a functionally identical view of the metadata gateway.**

Regarding claim 8, in contrast to the Examiner’s assertion, **Carre in view of Hamilton fails to teach or suggest wherein the metadata gateway is distributed over a plurality of servers, where each of the plurality of servers presents a functionally identical view of the metadata gateway.** The Examiner cites Fig. 1, the abstract, col. 3,

lines 5-67 and col. 4, lines 14-60 of Hamilton. However, the Examiner's reliance on Hamilton, even if combined with Carre, is misplaced.

Firstly, the cited passages do not describe or illustrate a metadata gateway distributed over a plurality of servers. To the contrary, Hamilton describes a system in which a *plurality of clients* sending *remote procedure calls* to a (single) server. For instance, FIG. 1 illustrates a plurality of client machines 22A – 22N interacting with a single server 22. Furthermore, Hamilton teaches that his system has “a large number of client computers connected to a transmission channel” and “generate method calls” and that a “server computer processes each method call received via the transmission channel” (col. 1, lines 56-65). Hamilton does not mention, nor illustrate, a metadata gateway distributed across a plurality of servers. The Examiner's cited passages regarding multiple clients issuing remote procedure calls to a single server (as illustrated in FIG. 1 of Hamilton) do not teach or suggest the specific limitation of a *metadata gateway distributed over a plurality of servers*. The remote procedure call transport mechanism taught by Hamilton is not a metadata gateway and thus cannot be said to teach or suggest, even if combined with Carre, a metadata gateway distributed over a plurality of servers.

### **Claims 10 and 13:**

**1. The cited art fails to teach or suggest a client generating a request to encode type information for an object, attribute, or event pertaining to management of one or more managed network objects, wherein each managed network object is a computer programming language object that represents one or more devices on a network.**

In regards to claim 10, contrary to the Examiner's contention, **Carre in view of Hamilton fails to teach or suggest a client generating a request to encode type information for an object, attribute, or event pertaining to management of one or more managed network objects, wherein each managed network object is a**

**computer programming language object that represents one or more devices on a network.** The Examiner cites the Abstract, FIG. 2A and 2B as well as column 3, lines 18-55 and column 5, lines 4-38 of Carre. The Examiner refers to Agent 1 of FIG. 3a, equating Agent 1 to the client of claim 10. However, Carre, even in combination with Hamilton, does not describe, either at the Examiner's cited passages or elsewhere, Agent 1 or any other client, *generating a request to encode type information* for an object, attribute, or event. As noted above regarding the rejection of claim 1, Carre is concerned with providing interface and address type conversions to allow objects specified according to different specification languages, such as ASN and IDL, to communicate and interact with each other by providing interfaces that allow non-CORBA objects to appear as CORBA objects to the outside world. Thus, Carre is concerned with automatic conversions as part of protocol translation during communication between two objects. Hamilton is also concerned with handling multiple protocols when making remote procedure calls.

Clients in Carre's system **do not generate requests to encode type information** for objects, attributes or events. Instead, client objects in Carre's system invoke services provided by server objects, via a standard CORBA interface (Carre, column 1, lines 9-19; column 1, line 59-column 2, line 46; column 5, lines 49-59). Carre's address conversion has nothing to with a client *generating a request to encode type information for an object, attribute or event*. Instead, Carre's address type conversion is performed as part of communicating with CMISE object that appear as CORBA objects. Nowhere does Carre mention a client *generating a request to encode type information*. Instead, Carre's interface unit converts object instances and object instance values into an IDL type to allow communication between CORBA objects and CMISE objects. Nowhere does Carre, even if combined with Hamilton, mention a client generating a request to encode type information for an object attribute or event.

**2. The cited art fails to teach or suggest sending the request to encode type information to an object request broker.**

In additional regards to claim 10, **Carre in view of Hamilton does not teach or suggest sending the request to encode type information to an object request broker.** The Examiner cites ORB and CMISE Gateway of FIG. 3a. However, Carre, even if combined with Hamilton, does not describe the ORB of FIG. 3a as receiving a request to encode type information from a client. Instead, Carre teaches object request brokers provide an infrastructure that enables objects to communicate in a distributed environment such that “it makes no difference” to the requesting objects in which computer system or in which form the target object is implemented (Carre, column 3, lines 56-63). Carre also teaches that a requesting object sends a request message to an object request broker and that the object request broker routes the request message to the target object (Carre, column 3, line 64-column 4, line 6). Thus, Carre teaches that object request brokers route request messages from a request object to the target object. Carre in view of Hamilton does not mention anything about object request brokers receiving requests to encode type information from a client.

**3. The cited art fails to teach or suggest a metadata gateway receiving the request to encode type information from the object request broker.**

In further regard to claim 10, **Carre in view of Hamilton also fails to teach or suggest a metadata gateway receiving the request to encode type information from the object request broker.** The Examiner admits that Carre fails to teach or suggest a metadata gateway receiving the request to encode type information from the object request broker and relies on Hamilton, citing Fig. 1, column 3, lines 4 – 67 and column 4, lines 14 – 60. As noted above, the cited passages describe how Hamilton’s client stubs pass remote procedure calls to client subcontract processes that, in turn, package the remote procedure calls for transport. On the server side, according to Hamilton, server subcontract processes passes the received package to server stubs that make procedure calls “to execute a specified method of the invoked object.” Thus, the cited portions of Hamilton describe a remote procedure call mechanism.



Nowhere does Hamilton, even in view of Carre, teach or suggest a metadata gateway receiving the request to encode type information from the object request broker. In fact, **nowhere does Hamilton mention anything about any sort of request to encode type information at all.** Hamilton's remote procedure call mechanism handles multiple, diverse, protocols, but does not, even when combined with Carre's remote procedure call mechanism, pertain to a metadata gateway receiving a request to encode type information from an object request broker.

**4. The cited art fails to teach or suggest storing the type information in a metadata repository, where the type information is stored in a database format in the metadata repository.**

In further regards to claim 10, **Carre in view of Hamilton further fails to teach or suggest storing the type information in a metadata repository, where the type information is stored in a database format in the metadata repository.** The Examiner does not cite any portion of Carre or Hamilton regarding this limitation of claim 10. In fact, the Examiner does not mention this limitation in the rejection of claim 10 at all.

The Examiner has previously cited CMISE/IDL of FIG. 3 and column 6, lines 10-35, specifically referring to Carre's conversion of address types from OSI types. However, these portions of Carre refer to the caching of structures, such as object instance values and object reference pairs for addresses already in an entity. Caching of object values and references is not the same as storing type information in a database format in a metadata repository. Additionally, as noted above regarding the rejection of claim 1, the CMISE/IDL interface unit of Carre is clearly a communication interface that allows non-CORBA objects to be interacted with via standard CORBA interfaces (Carre, column 5, lines 21-39). The CMISE/IDL interface unit is clearly not a metadata repository. Moreover, Carre does not describe or even mention anything regarding storing type information in the CMISE/IDL interface unit. Similarly, Hamilton also fails to teach or suggest storing type information in a metadata repository. Thus, Carre in

view of Hamilton fails to teach or suggest storing the type information in a metadata repository.

**Claim 11:**

**The cited art fails to teach or suggest translating the type information from the interface definition language to an abstract syntax notation and translating the type information from the abstract syntax notation to the database format.**

Regarding claim 11, Carre in view of Hamilton fails to teach or suggest translating the type information from the interface definition language to an abstract syntax notation and translating the type information from the abstract syntax notation to the database format. The Examiner cites column 1, lines 34-55 and column 5, line 60 to column 6, line 21, of Carre, specifically referring to Carre's use of semantic conversions. However, the cited passages, as well as the remainder of Carre, even if combined with Hamilton's teachings, only describe converting address values from abstract syntax notation to interface definition language and from interface definition language to abstract syntax notation. For example, Carre teaches that an OSI address value is converted to a correspondent IDL type (Carre, column 6, lines 1-2). Please note that Carre teaches that OSI objects are specified in ASN (Carre, column 1, lines 40-42). Thus, the Examiner's cited passages describe converting between ASN and IDL (and the reverse), but fail to mention anything about translating from an abstract syntax notation to a database format.

**Claim 12:**

**The cited art fails to teach or suggest translating the type information from the abstract syntax notation to an object specification language; and translating the type information from the object specification language to the interface definition language.**

Regarding claim 12, contrary to the Examiner's assertion, Carre in view of Hamilton does not teach or suggest **translating the type information from the abstract syntax notation to an object specification language** and **translating the type information from the object specification language to the interface definition language**. The Examiner cites column 1, lines 34-55 and column 5, line 60 – column 6, line 21, specifically referring to Carre's teachings regarding using semantic conversions. However, as noted above regarding claim 2, the cited passages, as well as the remainder of Carre, even if combined with Hamilton, only describe converting address values from abstract syntax notation to interface definition language as part of executing a remote procedure call.

For example, Carre teaches that an OSI address value is converted to a correspondent IDL type (Carre, column 6, lines 1-2). Please note that Carre teaches that OSI objects are specified in ASN (Carre, column 1, lines 40-42). Thus, the Examiner's cited passages describe converting directly between ASN and IDL (and the reverse), but fail to mention anything about translating from an abstract syntax notation to an object specification language and converting the object specification language to the IDL.

Additionally, Hamilton, even if combined with Carre, also fails to teach or suggest translating the type information from the abstract syntax notation to the object specification language and translating the type information from the object specification language to the interface definition language. Thus, Carre and Hamilton, whether considered singly or in combination, do not teach or suggest the limitations of claim 12.

#### **Claims 14, 20 and 21:**

**1. The cited art fails to teach or suggest a metadata repository comprising metadata concerning object classes for a plurality of managed objects.**

Regarding claim 14, contrary to the Examiner's assertion, **Carre in view of Hamilton fails to teach or suggest a metadata repository comprising metadata**

concerning object classes for a plurality of managed objects, wherein the metadata comprises information expressed in a database format, and wherein the managed objects are computer programming language objects corresponding to managed devices on a network. The Examiner cites CMISE/IDL of FIG. 3a as well as the abstract, FIGs 2A, 3A, column 3, lines 18-55 and column 5, lines 4-38 of Carre. However, as noted above, regarding the rejections of claim 1 and 10, the CMISE/IDL interface unit cited by the Examiner is clearly a communication interface that allows non-CORBA objects to be interacted with via standard CORBA interfaces (Carre, column 5, lines 21-39). **The CMISE/IDL interface unit is clearly not a metadata repository.** Moreover, Carre, even if combined with Hamilton, does not describe or even mention anything regarding storing metadata concerning object classes in the CMISE/IDL interface unit. Nowhere does Carre, even when combined with Hamilton, describe anything about storing metadata concerning object classes in a metadata repository.

**2. The cited art fails to teach or suggest a metadata gateway coupled to the metadata repository and to an object request broker, where the metadata gateway provides translation of the metadata to and from the database format and an interface definition language**

In further regard to claim 14, Carre in view of Hamilton also fails to teach or suggest a metadata gateway coupled to the metadata repository and to an object request broker, where the metadata gateway is operable to send and receive metadata from the database, where the metadata gateway provides translation of the metadata to and from the database format and an interface definition language.

The Examiner admits that Carre fails to teach this limitation of claim 14 and relies on Hamilton, again citing Fig. 1, column 3, lines 4 – 67 and column 4, lines 14 – 60. As noted above, the cited passages describe how Hamilton's client stubs pass remote procedure calls to client subcontract processes that, in turn, package the remote procedure calls for transport. On the server side, according to Hamilton, server subcontract processes passes the received package to server stubs that make procedure calls "to

execute a specified method of the invoked object.” Thus, the cited portions of Hamilton describe a remote procedure call mechanism.

However, Hamilton’s remote procedure call mechanism, even if combined with Carre’s teachings, does not involve, nor pertain to, a metadata gateway that provides translation of metadata to and from a database format and an interface definition language. Instead, Hamilton teaches that protocol-dependent values that match a method descriptor are located in a database and passed to a server stub that executes the method indicated by the method descriptor (Hamilton column 1, line 56 – column 2, line 34; FIG. 3; column 5, lines 20-39; and column 6, lines 23-45). Please note that Hamilton does not teach that any translation of metadata to and from a database format and an interface definition language. Instead, Hamilton, even when combined with Carre, teaches that values (e.g., protocol-dependent values) are located in a database and passed without translation to server stub processes for use when executing an indicated method. Nothing about Hamilton’s system involves a metadata gateway providing translation of metadata to and from a database format and an interface definition language.

#### **Claims 17-19:**

**The cited art fails to teach or suggest a metadata gateway that includes a library of data types expressed in an abstract syntax notation, a plurality of object types, wherein each object type includes one or more of the data types from the library of data types.**

Regarding claim 17, Carre in view of Hamilton fails to teach or suggest wherein the metadata gateway includes a library of data types expressed in an abstract syntax notation, a plurality of object types, wherein each object type includes one or more of the data types from the library of data types. The Examiner cites, FIG. 2A, 3A, column 4, lines 7-62 and column 5, line 39 to column 6, line 35 of Carre. However, the Examiner has failed to cite any portion of Carre (or Hamilton) that describes a library of data types *expressed in an abstract syntax notation*. Additionally,

Carre's CMISE Gateway, which the Examiner equates to the metadata gateway of Appellants' claims, does not include a library of data types. Carre makes no mention of his CMISE Gateway including a library of data types expressed in an abstract syntax notation. Instead, Carre describes his CMISE Gateway as providing address type conversion between two different object interface specifications, such as between IDL and C++. Merely providing address type conversion does not imply including a library of data types and a plurality of object type, each including one or more of the data types from the library. The Examiner's interpretation of Carre's CMISE Gateway is incorrect.

Additionally, Carre in view of Hamilton also fails to teach or suggest wherein the metadata gateway includes an interface to the plurality of object types, wherein the interface is operable to provide one or more clients with access to the metadata as expressed in the interface definition language. The Examiner's cited passages of Carre do not describe the CMISE Gateway, which the Examiner equates to the metadata gateway of Appellants' claims, as including an interface to a plurality of object types, where the interface provides clients with access to the metadata. Instead, the Examiner's cited passages teach that Carre's CMISE Gateway translates objects and object address types from ASN to IDL and from IDL to ASN. Nowhere does Carre describe that the CMISE Gateway includes an interface providing clients access to metadata.

Hamilton also fails to describe or teach anything that overcomes Carre's failure to teach the subject matter on which the Examiner relies. Thus, Carre and Hamilton, whether considered singly or in combination, do not teach or suggest a metadata gateway that includes a library of data types expressed in an abstract syntax notation, a plurality of object types, wherein each object type includes one or more of the data types from the library of data types. Carre and Hamilton, whether considered singly or in combination, also fail to teach or suggest that the metadata gateway includes an interface to the plurality of object types, wherein the interface is operable to provide one or more clients with access to the metadata as expressed in the interface definition language.

**Claims 22 and 26:**

1. The cited art fails to teach or suggest a client generating a request for type information for an attribute or event pertaining to management of one or more managed network objects.

In regards to claim 22, contrary to the Examiner's contention, **Carre in view of Hamilton fails to teach or suggest a client generating a request for type information for an attribute or event pertaining to management of one or more managed network objects, wherein each managed network object is a computer programming language object representing one or more devices on a network.** As argued above regarding claim 1, the Examiner erroneously equates any address conversion performed as part of any remote procedure call with specifically *generating a request for type information* for an attribute or event pertaining to management of one or more managed network objects. However, as shown below, merely performing an address conversion does not teach or suggest a *client generating a request* for type information.

Carre pertains to address conversion between CORBA objects and OSI objects (Carre - col. 1, lines 9-19; col. 1, line 59 - col. 2, line 46) and to the transforming of object interfaces column 5, lines 49-59), while Hamilton teaches a system for handling diverse protocols with remote procedure calls (Abstract, col. 1, line 57 - col. 2, line 34). The Examiner cites the Abstract, FIG. 2A, 2B, as well as column 3, line 18 to column 4, line 62 and column 6, lines 10-35 of Carre.

Carre describes how objects interact and communicate using a CORBA infrastructure. In order to permit address interaction between objects that use different addressing modes, Carre's system converts an address type with an address value according to one addressing mode to a corresponding type of another specification language or addressing mode. (Carre, column 1, line 59 - column 2, line 6; column 2, lines 11-14; lines 25-28; and lines 34- 39). Thus, Carre is concerned with *converting address types and object interfaces*, but fails to teach anything regarding a client

*generating a request for type information*. Even if combined with Hamilton, Carre's system does not include a client *generating a request for type information*, either at the Examiner's cited passages or elsewhere.

As noted above, the Examiner apparently equates any address conversion that may occur during the course of a remote procedure as *a client generating a request for type information*. However, Address conversions made during the normal course of other processing do not include or require *a client generating a request for type information*. Carre in view of Hamilton teaches address conversion, but fails to teach or suggest a client generating a request for type information. Converting address types and object interfaces as part of a protocol conversion, as taught by Carre in view of Hamilton, does not include *a client generating a request for type information*. Please see the rejection of claim 1 above for a more detailed discussion of Carre's failure (even when considered in combination with Hamilton) to teach or suggest a client generating a request for type information.

Moreover, the Examiner has failed to consider the fact that Carre's clients request specific services from particular server objects and that Carre's system automatically make type and address conversions between different object specification languages. For instance, Carre's client objects request services provided by service objects by sending a request message to a service object that contains an operation, a target object, one or more parameters, and, optionally, a request context (Carre, column 3, lines 47-53).

However, as argued above and previously **Clients in Carre's system do not generate requests for type information**, but instead request a specific service from a particular server object. Carre's system may automatically make any type and address conversions between different object specification languages necessarily to allow the client and server object to communicate. Thus, in a system resulting from the Examiner's combination of Carre and Hamilton, type and address conversion may take place, **without the agent (client) knowing** that such conversions are performed on the



parameters, data, and addresses of various requests between Carre's manager and agent objects.

Furthermore, there is no need in Carre's system for a *client to generate a request* for type information for an attribute or event. In fact, the purpose of Carre's teachings appears to be to perform the address type conversion for the client **so that the client does not have to modify its own interface in order to communicate with an object employing a different addressing mode**. See, e.g., col. 1, line 43 – col. 2, line 6. **Therefore, Carre actually teaches away from a client generating a request for type information for an attribute or event.**

**2. The cited art fails to teach or suggest sending the request for type information generated by a client to an object request broker.**

In further regard to claim 22, Carre in view of Hamilton also fails to disclose ***sending the request for type information generated by a client to an object request broker***. The Examiner admits that Carre fails to teach or suggest sending a request for type information to an object request broker and relies on Hamilton, repeating the citations from the rejection (Hamilton, Fig. 1, abstract, column 3, lines 4 – 67 and column 4, lines 14 – 60). However, like Carre, Hamilton does not teach or suggest anything about sending *a request for type information* to an object request broker. Hamilton teaches a system for handling diverse protocols with remote procedure calls. Specifically, Hamilton teaches that method descriptors located within method calls received by a server are specified in a protocol-dependent format. Hamilton also teaches that the method descriptors are compared to other protocol-dependent values and matching values are passed to a protocol-independent processing module that executes the method and returns the reply to the client. See, e.g., Hamilton, column 1, line 57-column 2, line 34).

Appellants have previously argued that data marshalling and method invocations may involve address conversion, but do not involve sending a request (generated by a

client) for type information to an object request broker. In the Response to Arguments, the Examiner again refers to Hamilton's subcontract server 58 "performing data marshalling and other operations of method invocations," without providing any addition response to Appellants' argument. The Examiner specifically refers to "managing operations such as OSI objects on the Agent upon requests from the Manager unit" and cites item M of Fig. 2a from Carre. However, Carre, even if combined with Hamilton, does not describe his manager unit as sending (or receiving) a request (generated by a client) for type information. As noted previously, the Examiner is erroneously equating the automatic type and conversions performed as part of a protocol translation with the specific limitation of a client generating a request for type information.

**3. The cited art fails to teach or suggest the client receiving the translated type information.**

**In further regard to claim 22, Carre in view of Hamilton also fails to teach or suggest the client receiving the translated type information.** Claim 22 requires, among other things, that the client generate a request for type information for an attribute or event, and that the client receive the translated type information for the attribute or event. Following the Examiner's line of reasoning, agent (A) or manager unit (U) would have to receive the requested type information, translated to IDL. However, the automatic address and/or type conversions performed as part of Carre's protocol translation are performed on communication message sent between Carre's agent and manager units. Carre's automatic conversions are preformed so that an object configured to communicate with one protocol can understand messages from another object configured to communicate using a different protocol. **Any address or other information converted or translated by Carre's system is not returned to the object or unit sending the request.** Instead, the converted data, parameters and address are sent to the receiving object. Any return communication from the receiving object back to the requesting object may also be subject to Carre's protocol translation and therefore Carre's automatic address conversions.

Thus, the converted address and/or other information converted as part of delivering a request from Carre's managing object (M) to Carre's agent object (A) is not then returned to the managing object. Instead, a message sent from M to A in Carre's system would be converted as part of Carre's protocol translation and then sent to A. The converted message is not returned to M. Instead, a separate, return, message from A may be sent back to M and may also be translated to a different protocol. Thus, Carre's object communication messages, even if combined with Hamilton, cannot be considered request for type information, as recited in claim 22.

**4. The cited art fails to teach or suggest a metadata gateway receiving the request for type information from the object request broker.**

In further regard to claim 22, **Carre in view of Hamilton further fails to disclose a metadata gateway receiving the request for type information from the object request broker.** The Examiner admits that Carre fails to teach or suggest a metadata gateway receiving the request for type information from the object request broker and relies on Hamilton. However, Hamilton, even when combined with Carre, fails to mention anything regarding a metadata gateway *receiving a request for type information* from an object request broker. The Examiner, in the response to arguments, again cites Fig. 1, column 3, lines 4 – 67 and column 4, lines 14 – 60 of Hamilton. However, the cited portions do not describe anything regarding a request for type information or about a metadata gateway receiving a request for type information from an object request broker. Instead, the cited passages describe how Hamilton's client stubs pass remote procedure calls to client subcontract processes that, in turn, package the remote procedure calls for transport. On the server side, according to Hamilton, server subcontract processes passes the received package to server stubs that make procedure calls "to execute a specified method of the invoked object." Thus, the cited portions of Hamilton describe a remote procedure call mechanism. Nowhere does Hamilton mention anything regarding a receiving a request for type information from an object request broker. Thus, contrary to the Examiner's assertion, Hamilton combined with Carre does

not teach or suggest **a metadata gateway receiving the request for type information from the object request broker.**

**5. The cited art fails to teach or suggest reading the type information from a metadata repository, where the type information is stored in a database format.**

Contrary to the Examiner's contention, **Carre in view of Hamilton does not teach or suggest reading the type information from a metadata repository, wherein the type information is stored in a database format in the metadata repository.** The Examiner cites CMISE/IDL FIG. 3. However, the cited CMISE/IDL is not a metadata repository from which type information is read. In response, the Examiner refers to "using CMISE/IDL fig. 3 as a metadata repository to manage/store the OSI objects translated from type conversion" and also cites column 6, lines 1 – 35 of Carre. Column 6, lines 1 – 35 describe the conversion of ASN types to a correspondent IDL type. However, Carre teaches that the CMISE/IDL is an "interface unit" that contains the "CMISE object and the services assigned to this object". Carre also teaches that the CMISE object is specified by an IDL interface and acts, and thus appears, like a CORBA object (Carre, column 5, lines 21-39). Thus, the CMISE/IDL cited by the Examiner is a communication interface that allows non-CORBA objects to be interacted with via standard CORBA interfaces. The CMISE/IDL interface unit is clearly not a metadata repository. Moreover, Appellants' claim specifically recites that type information is stored *in a database format* in the metadata repository. Carre's gateway does not store type information in a database format and thus cannot be considered the metadata repository of Appellants' claim.

**6. The cited art fails to teach or suggest the client receiving the translated type information for the attribute or event through the object request broker.**

**Carre in view of Hamilton also fails to teach or suggest the client receiving the translated type information for the attribute or event through the object request broker.** The Examiner cites column 6, lines 10-35 of Carre. However, this portion of Carre does not describe a client receiving the translated type information for the attribute or event through the object request broker. Instead, as noted above, the cited passage is part of a description of converting objects from ASN to IDL and vice versa as well as caching of converting object instances. As described above, converting CMISE and CORBA objects is not the same as client generating a request to type information, and receiving the translated type information for the attribute or event through an object request broker. The Examiner's cited passage makes no mention of any client receiving translated type information for an attribute or event through an object request broker.

Moreover, clients in Carre's system, even if combined with Hamilton's teachings, do not receive any translated type information through an object request broker. Carre and Hamilton both teach remote procedure call mechanism for executing remote methods over a network. Consequently, clients in Carre's and Hamilton's systems, whether considered singly or in combination, receive the results of the executed method. Clients in Carre and Hamilton do not receive translated type information through an object request broker.

**Claims 23 and 24:**

**1. The cited art fails to teach or suggest translating the type information from the database format to an abstract syntax notation and translating the type information from the abstract syntax notation to the interface definition language.**

Regarding claim 23, contrary to the Examiner's assertion, Carre in view of Hamilton does not teach or suggest **translating the type information from the database format to an abstract syntax notation** and **translating the type information from the abstract syntax notation to the interface definition language**. The Examiner cites column 1, lines 34-55 and column 5, line 60 – column 6, line 21, specifically

referring to Carre's teachings regarding using semantic conversions. However, the cited passages, as well as the remainder of Carre, even if combined with Hamilton, only describe converting address values from abstract syntax notation to interface definition language as part of executing a remote procedure call. For example, Carre teaches that an OSI address value is converted to a correspondent IDL type (Carre, column 6, lines 1-2). Please note that Carre teaches that OSI objects are specified in ASN (Carre, column 1, lines 40-42). Thus, the Examiner's cited passages describe converting between ASN and IDL (and the reverse), but fail to mention anything about translating from a database format to an abstract syntax notation.

Additionally, Hamilton, even if combined with Carre, also fails to teach or suggest translating the type information from the database format to an abstract syntax notation and translating the type information from the abstract syntax notation to the interface definition language. Thus, Carre and Hamilton, whether considered singly or in combination, do not teach or suggest the limitations of claim 23.

Please see the discussion of claim 2 above for a more detailed discussion of Carre and Hamilton's failure to teach or suggest translating the type information from the database format to an abstract syntax notation and translating the type information from the abstract syntax notation to the interface definition language, as recited in Appellants' claim.

**Claim 25:**

**The cited art fails to teach or suggest translating the type information from the abstract syntax notation to an object specification language; and translating the type information from the object specification language to the interface definition language.**

Regarding claim 25, contrary to the Examiner's assertion, Carre in view of Hamilton does not teach or suggest **translating the type information from the abstract**

syntax notation to an object specification language and translating the type information from the object specification language to the interface definition language. The Examiner cites column 1, lines 34-55 and column 5, line 60 – column 6, line 21, specifically referring to Carre's teachings regarding using semantic conversions. However, as noted above regarding claim 2, the cited passages, as well as the remainder of Carre, even if combined with Hamilton, only describe converting address values from abstract syntax notation to interface definition language as part of executing a remote procedure call.

For example, Carre teaches that an OSI address value is converted to a correspondent IDL type (Carre, column 6, lines 1-2). Please note that Carre teaches that OSI objects are specified in ASN (Carre, column 1, lines 40-42). Thus, the Examiner's cited passages describe converting directly between ASN and IDL (and the reverse), but fail to mention anything about translating from an abstract syntax notation to an object specification language and converting the object specification language to the IDL.

Additionally, Hamilton, even if combined with Carre, also fails to teach or suggest translating the type information from the abstract syntax notation to the object specification language and translating the type information from the object specification language to the interface definition language. Thus, Carre and Hamilton, whether considered singly or in combination, do not teach or suggest the limitations of claim 25.

#### **Claims 27 and 31:**

1. The cited art fails to teach or suggest a client generating a request to encode type information for an object, attribute, or event pertaining to management of one or more managed network objects, wherein each managed network object is a computer programming language object that represents one or more devices on a network.

In regards to claim 27, contrary to the Examiner's contention, **Carre in view of Hamilton fails to teach or suggest a client generating a request to encode type information for an object, attribute, or event pertaining to management of one or more managed network objects, wherein each managed network object is a computer programming language object that represents one or more devices on a network.** The Examiner cites the Abstract, FIG. 2A and 2B as well as column 3, lines 18-55 and column 5, lines 4-38 of Carre. The Examiner refers to Agent 1 of FIG. 3a, equating Agent 1 to the client of claim 10. However, Carre, even in combination with Hamilton, does not describe, either at the Examiner's cited passages or elsewhere, Agent 1 or any other client, *generating a request to encode type information* for an object, attribute, or event. As noted above regarding the rejections of claims 1 and 10, Carre is concerned with providing interface and address type conversions to allow objects specified according to different specification languages, such as ASN and IDL, to communicate and interact with each other by providing interfaces that allow non-CORBA objects to appear as CORBA objects to the outside world. Thus, Carre is concerned with automatic conversions as part of protocol translation during communication between two objects. Hamilton is also concerned with handling multiple protocols when making remote procedure calls.

Clients in Carre's system **do not generate requests to encode type information** for objects, attributes or events. Instead, client objects in Carre's system invoke services provided by server objects, via a standard CORBA interface (Carre, column 1, lines 9-19; column 1, line 59-column 2, line 46; column 5, lines 49-59). Carre's address conversion has nothing to with a client *generating a request to encode type information for an object, attribute or event*. Instead, Carre's address type conversion is performed as part of communicating with CMISE object that appear as CORBA objects. Nowhere does Carre mention a client *generating a request to encode type information*. Instead, Carre's interface unit converts object instances and object instance values into an IDL type to allow communication between CORBA objects and CMISE objects. Nowhere does Carre, even if combined with Hamilton, mention a client generating a request to encode type information for an object attribute or event.



**2. The cited art fails to teach or suggest sending the request to encode type information to an object request broker.**

In additional regards to claim 27, **Carre in view of Hamilton does not teach or suggest sending the request to encode type information to an object request broker.** The Examiner cites ORB and CMISE Gateway of FIG. 3a. However, Carre, even if combined with Hamilton, does not describe the ORB of FIG. 3a as receiving a request to encode type information from a client. Instead, Carre teaches object request brokers provide an infrastructure that enables objects to communicate in a distributed environment such that “it makes no difference” to the requesting objects in which computer system or in which form the target object is implemented (Carre, column 3, lines 56-63). Carre also teaches that a requesting object sends a request message to an object request broker and that the object request broker routes the request message to the target object (Carre, column 3, line 64-column 4, line 6). Thus, Carre teaches that object request brokers route request messages from a request object to the target object. Carre in view of Hamilton does not mention anything about object request brokers receiving requests to encode type information from a client.

**3. The cited art fails to teach or suggest a metadata gateway receiving the request to encode type information from the object request broker.**

**Carre in view of Hamilton also fails to teach or suggest a metadata gateway receiving the request to encode type information from the object request broker.** The Examiner admits that Carre fails to teach or suggest a metadata gateway receiving the request to encode type information from the object request broker and relies on Hamilton, citing Fig. 1, column 3, lines 4 – 67 and column 4, lines 14 – 60. As noted above, the cited passages describe how Hamilton’s client stubs pass remote procedure calls to client subcontract processes that, in turn, package the remote procedure calls for transport. On the server side, according to Hamilton, server subcontract processes passes the received package to server stubs that make procedure calls “to execute a specified

method of the invoked object.” Thus, the cited portions of Hamilton describe a remote procedure call mechanism.

Nowhere does Hamilton, even in view of Carre, teach or suggest a metadata gateway receiving the request to encode type information from the object request broker. In fact, **nowhere does Hamilton mention anything about any sort of request to encode type information at all.** Hamilton’s remote procedure call mechanism handles multiple, diverse, protocols, but does not, even when combined with Carre’s remote procedure call mechanism, pertain to a metadata gateway receiving a request to encode type information from an object request broker.

**4. The cited art fails to teach or suggest storing the type information in a metadata repository, where the type information is stored in a database format in the metadata repository.**

In further regards to claim 27, **Carre in view of Hamilton further fails to teach or suggest storing the type information in a metadata repository, where the type information is stored in a database format in the metadata repository.** The Examiner has cited column 6, lines 1-35, specifically referring to Carre’s conversion of address types from OSI types. However, these portions of Carre refer to the caching of structures, such as object instance values and object reference pairs for addresses already in an entity. Caching of object values and references is not the same as storing type information in a database format in a metadata repository. Additionally, as noted above regarding the rejection of claim 1, the CMISE/IDL interface unit of Carre is clearly a communication interface that allows non-CORBA objects to be interacted with via standard CORBA interfaces (Carre, column 5, lines 21-39). The CMISE/IDL interface unit is clearly not a metadata repository. Moreover, Carre does not describe or even mention anything regarding storing type information in the CMISE/IDL interface unit. Similarly, Hamilton also fails to teach or suggest storing type information in a metadata repository. Thus, Carre in view of Hamilton fails to teach or suggest storing the type information in a metadata repository.

**Claims 28 and 29:**

**The cited art fails to teach or suggest translating the type information from the interface definition language to an abstract syntax notation and translating the type information from the abstract syntax notation to the database format.**

Regarding claim 28, **Carre in view of Hamilton fails to teach or suggest translating the type information from the interface definition language to an abstract syntax notation and translating the type information from the abstract syntax notation to the database format.** The Examiner cites column 1, lines 34-55 and column 5, line 60 to column 6, line 21, of Carre, specifically referring to Carre's use of semantic conversions. However, the cited passages, as well as the remainder of Carre, even if combined with Hamilton's teachings, only describe converting address values from abstract syntax notation to interface definition language and from interface definition language to abstract syntax notation. For example, Carre teaches that an OSI address value is converted to a correspondent IDL type (Carre, column 6, lines 1-2). Please note that Carre teaches that OSI objects are specified in ASN (Carre, column 1, lines 40-42). Thus, the Examiner's cited passages describe converting between ASN and IDL (and the reverse), but fail to mention anything about translating from an abstract syntax notation to a database format.

**Claim 30:**

**The cited art fails to teach or suggest translating the type information from the abstract syntax notation to an object specification language; and translating the type information from the object specification language to the interface definition language.**

Regarding claim 30, contrary to the Examiner's assertion, Carre in view of Hamilton does not teach or suggest **translating the type information from the abstract**

**syntax notation to an object specification language and translating the type information from the object specification language to the interface definition language.** The Examiner cites column 1, lines 34-55 and column 5, line 60 – column 6, line 21, specifically referring to Carre’s teachings regarding using semantic conversions. However, as noted above regarding claim 2, the cited passages, as well as the remainder of Carre, even if combined with Hamilton, only describe converting address values from abstract syntax notation to interface definition language as part of executing a remote procedure call.

For example, Carre teaches that an OSI address value is converted to a correspondent IDL type (Carre, column 6, lines 1-2). Please note that Carre teaches that OSI objects are specified in ASN (Carre, column 1, lines 40-42). Thus, the Examiner’s cited passages describe converting directly between ASN and IDL (and the reverse), but fail to mention anything about translating from an abstract syntax notation to an object specification language and converting the object specification language to the IDL.

Additionally, Hamilton, even if combined with Carre, also fails to teach or suggest translating the type information from the abstract syntax notation to the object specification language and translating the type information from the object specification language to the interface definition language. Thus, Carre and Hamilton, whether considered singly or in combination, do not teach or suggest the limitations of claim 30.

### **Second Ground of Rejection:**

Claims 6, 13, 15 and 16 stand finally rejected 35 U.S.C. § 103(a) as being unpatentable over Carre in view of Hamilton and further in view of Kung et al. (U.S. Patent 6,775,267) (hereinafter “Kung”). Appellants respectfully traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

**Claim 6:**

Appellants traverse the rejection of claim 6 and assert that claim 6 is patentable for at least the reasons presented above regarding its independent claim.

**Claim 13:**

Appellants traverse the rejection of claim 13 and assert that claim 13 is patentable for at least the reasons presented above regarding its independent claim.

**Claim 15:**

**The cited art fails to teach or suggest where the managed devices comprise a telephone system.**

In regards to claim 15, contrary to the Examiner's assertion, Carre in view of Hamilton in further view of Kung **does not teach or suggest that the managed devices comprise a telephone system.** The Examiner cites figs. 1, 2, the abstract, col. 4 lines 2-59 and col. 11, line 66 – col. 12, line 51 of Kung. However, the Examiner's reliance on Kung is misplaced. Kung does not teach where the managed objects include a telephone network. Appellants' claim recites that the *managed* devices on the network include a telephone system. In contrast, Kung, whether considered singly or in combination teaches communication over a telephone network not that managed devices include a telephone system. For instance, Kung, at the Examiner's cited passage teaches, "the broadband residential gateway 300 may provide one or more telephone port connections (for example, plain old telephone system), Ethernet connections, coaxial connections, fiber distributed data interface (FDDI) connections, wireless local area network (LAN) connections ..." (Kung, col. 4, lines32-59).

Thus, Kung teaches a broadband residential gateway that may be able to communicate over any of various types of communications channels (e.g., POTS,

Ethernet, LAN, FDDI, etc), but teaches communication over a telephone system is very different from including a telephone network as a managed device, as recited in Appellants' claim. The fact that a system may communicate over a particular communications channel does not imply that the communication channel is a managed device on a network, contrary to the Examiner's contention.

Furthermore, Carre and Hamilton, even if combined with Kung, also fail to teach or suggest where the managed devices include a telephone system. Thus, the Examiner's rejection of claim 15 is not supported by the cited art and removal thereof is respectfully requested.

**Claim 16:**

**The cited art fails to teach or suggest where the managed devices comprise a network switch.**

In regards to claim 16, contrary to the Examiner's assertion, Carre in view of Hamilton in further view of Kung **does not teach or suggest that the managed devices comprise a network switch**. The Examiner cites figs. 1, 2, the abstract, col. 4 lines 2-59 and col. 11, line 66 – col. 12, line 51 of Kung. However, the Examiner's reliance on Jung is misplaced. Kung does not teach where the managed objects include a network switch. Appellants' claim recites that the *managed* devices on the network include a network switch. In contrast, Kung, whether considered singly or in combination teaches communication over a network and therefore via a network switch, not that managed devices include a network switch. For instance, Kung, at the Examiner's cited passage teaches, "the broadband residential gateway 300 may provide one or more telephone port connections (for example, plain old telephone system), Ethernet connections, coaxial connections, fiber distributed data interface (FDDI) connections, wireless local area network (LAN) connections ..." (Kung, col. 4, lines 32-59).

Thus, Kung teaches a broadband residential gateway that may be able to communicate over any of various types of communications channels (e.g., Ethernet, LAN, FDDI, etc), but teaches communication over a network switch is very different from including a network switch in a group of managed devices, as recited in Appellants' claim. The fact that a system may communicate over a particular communications channel does not imply that the communication channel **is a managed device** on a network, contrary to the Examiner's contention.

Furthermore, Carre and Hamilton, even if combined with Kung, also fail to teach or suggest where the managed devices include a network switch. Thus, the Examiner's rejection of claim 16 is not supported by the cited art and removal thereof is respectfully requested.

## **CONCLUSION**

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-31 was erroneous, and reversal of his decision is respectfully requested.

**Since Appellants' previous appeal (Brief filed December 28, 2004) was not heard by the Board, no fee should be due for this Appeal Brief (see M.P.E.P. § 1207.04). Also, no extension of time is due since this Appeal Brief is filed within one month of the mailing date of the Notice of Panel Decision.** The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-46200/RCK.

Respectfully submitted,

/Robert C. Kowert/

Robert C. Kowert, Reg. #39,255  
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
(512) 853-8850

Date: December 13, 2007



## **VIII. CLAIMS APPENDIX**

The claims on appeal are as follows.

1. A method for managing a network, the method comprising:

a client generating a request for type information for an attribute or event pertaining to management of one or more managed network objects, wherein the request is expressed in an interface definition language, wherein the interface definition language is operable to define object interfaces across a plurality of platforms and across a plurality of programming languages, wherein each managed network object is a computer programming language object representing one or more devices on a network;

sending the request for type information to an object request broker;

a metadata gateway receiving the request for type information from the object request broker;

reading the type information from a metadata repository, wherein the type information is stored in a database format in the metadata repository;

translating the type information from the database format to the interface definition language;

the metadata gateway sending the translated type information to the object request broker; and

the client receiving the translated type information for the attribute or event through the object request broker, wherein the translated type information is expressed in the interface definition language.

2. The method of claim 1, wherein the translating the type information from the database format to the interface definition language comprises:

translating the type information from the database format to an abstract syntax notation; and

translating the type information from the abstract syntax notation to the interface definition language.

3. The method of claim 2, wherein the abstract syntax notation is Abstract Syntax Notation One (ASN1).

4. The method of claim 2, wherein the translating the type information from the abstract syntax notation to the interface definition language comprises:

translating the type information from the abstract syntax notation to an object specification language; and

translating the type information from the object specification language to the interface definition language.

5. The method of claim 1, wherein the sending the request for type information to an object request broker, the metadata gateway receiving the request for type information from the object request broker, the metadata gateway sending the translated type information to the object request broker, and the client receiving the translated type information for the attribute or event through the object request broker are effected via an internet inter-object communication protocol.

6. The method of claim 5, wherein the internet inter-object communication protocol comprises Internet Inter-Object Protocol (IIOP).

7. The method of claim 1, wherein the metadata gateway is implemented on a single server computer system.

8. The method of claim 1, wherein the metadata gateway is distributed over a plurality of servers, wherein each of the plurality of servers presents a functionally identical view of the metadata gateway.

9. The method of claim 1, wherein the interface definition language is class-independent.

10. A method for managing a network, the method comprising:

a client generating a request to encode type information for an object, attribute, or event pertaining to management of one or more managed network objects, wherein the request is expressed in an interface definition language, wherein the interface definition language is operable to define object interfaces across a plurality of platforms and across a plurality of programming languages, wherein each managed network object is a computer programming language object that represents one or more devices on a network;

sending the request to encode the type information to an object request broker;

a metadata gateway receiving the request to encode the type information from the object request broker;

translating the type information from the interface definition language to a database format; and

storing the type information in a metadata repository, wherein the type information is stored in a database format in the metadata repository.

11. The method of claim 10, wherein the translating the type information from the interface definition language to the database format comprises:

translating the type information from the interface definition language to an abstract syntax notation; and

translating the type information from the abstract syntax notation to the database format.

12. The method of claim 11, wherein the translating the type information from the interface definition language to the abstract syntax notation comprises:

translating the type information from the interface definition language to an object specification language; and

translating the type information from the object specification language to the abstract syntax notation.

13. The method of claim 10, wherein the sending the request to an object request broker and the metadata gateway receiving the request to encode the type information from the object request broker are effected via an internet inter-object communication protocol.

14. A network management system comprising:

a metadata repository, wherein the metadata repository comprises metadata concerning object classes for a plurality of managed objects, wherein the metadata comprises information expressed in a database format, and wherein the managed objects are computer programming language objects corresponding to managed devices on a network; and

a metadata gateway which is communicatively coupled to the metadata repository and to an object request broker, wherein the metadata gateway is operable to send and receive the metadata from the database, wherein the metadata gateway provides translation of the metadata to and from the database format and an interface definition language, wherein the interface definition language is operable to define object interfaces across a plurality of platforms and across a plurality of programming languages.

15. The network management system of claim 14, wherein the managed devices comprise a telephone system.

16. The network management system of claim 14, wherein the managed devices comprise a network switch.

17. The network management system of claim 14, wherein the metadata gateway further comprises:

a library of data types expressed in an abstract syntax notation, wherein the abstract syntax notation comprises a metadata notation language;

a plurality of object types, wherein each object type comprises one or more of the data types from the library of data types; and

an interface to the plurality of object types, wherein the interface is operable to provide one or more clients with access to the metadata as expressed in the interface definition language.

18. The network management system of claim 17, wherein the interface to the plurality of object types is a programming-language-independent and platform-independent interface.

19. The network management system of claim 17, wherein the plurality of object types comprise CORBA objects.

20. The network management system of claim 14, wherein the object request broker is configurable to be accessed by a plurality of network management clients to obtain the metadata as expressed in the generic interface.

21. The network management system of claim 14, wherein the object request broker comprises a CORBA ORB.

22. A tangible, computer-readable storage medium comprising program instructions, wherein the program instructions are computer-executable to implement:

a metadata gateway receiving a request for type information from an object request broker, wherein the type information pertains to management of one or more managed network objects, wherein each managed network object is a computer programming language object that represents one or more devices on a network;

reading the type information from a metadata repository, wherein the type information is stored in a database format in the metadata repository;

translating the type information from the database format to an interface definition language; and

the metadata gateway sending the translated type information to the object request broker.

23. The computer-readable storage medium of claim 22, wherein in translating the type information from the database format to the interface definition language, the program instructions are further computer-executable to implement:

translating the type information from the database format to an abstract syntax notation; and

translating the type information from the abstract syntax notation to the interface definition language.

24. The computer-readable storage medium of claim 23, wherein the abstract syntax notation is Abstract Syntax Notation One (ASN1).

25. The computer-readable storage medium of claim 22, wherein in translating the type information from the abstract syntax notation to the interface definition language, the program instructions are further computer-executable to implement:

translating the type information from the abstract syntax notation to an object specification language; and

translating the type information from the object specification language to the interface definition language.

26. The computer accessible medium of claim 22, wherein the interface

definition language is class-independent.

27. A tangible, computer-readable storage medium comprising program instructions which are computer-executable to implement:

a metadata gateway receiving a request to encode type information from an object request broker, wherein the type information pertains to management of one or more managed network objects;

translating the type information from an interface definition language to a database format; and

storing the type information in a metadata repository, wherein the type information is stored in a database format in the metadata repository.

28. The computer-readable storage medium of claim 27, wherein in translating the type information from the interface definition language to the database format, the program instructions are further computer-executable to implement:

translating the type information from the interface definition language to an abstract syntax notation; and

translating the type information from the abstract syntax notation to the database format.

29. The computer-readable storage medium of claim 28, wherein the abstract syntax notation is Abstract Syntax Notation One (ASN1).

30. The computer-readable storage medium of claim 27, wherein in translating the type information from the interface definition language to the abstract syntax notation, the program instructions are further computer-executable to implement:



translating the type information from the interface definition language to an object specification language; and

translating the type information from the object specification language to the abstract syntax notation.

31. The computer-readable storage medium of claim 27, wherein the interface definition language is class-independent.

## **IX. EVIDENCE APPENDIX**

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

**X.     RELATED PROCEEDINGS APPENDIX**

There are no related proceedings.